

PRIOR ART

FIG. 1

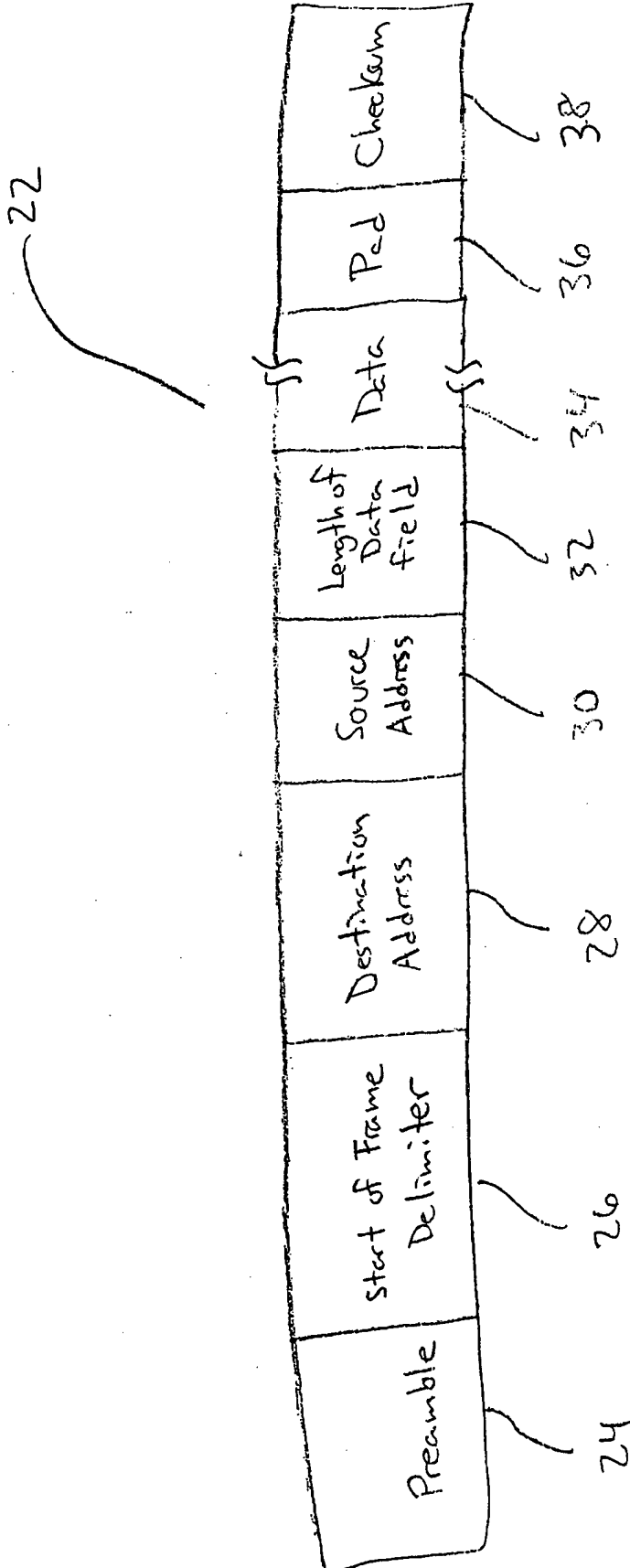
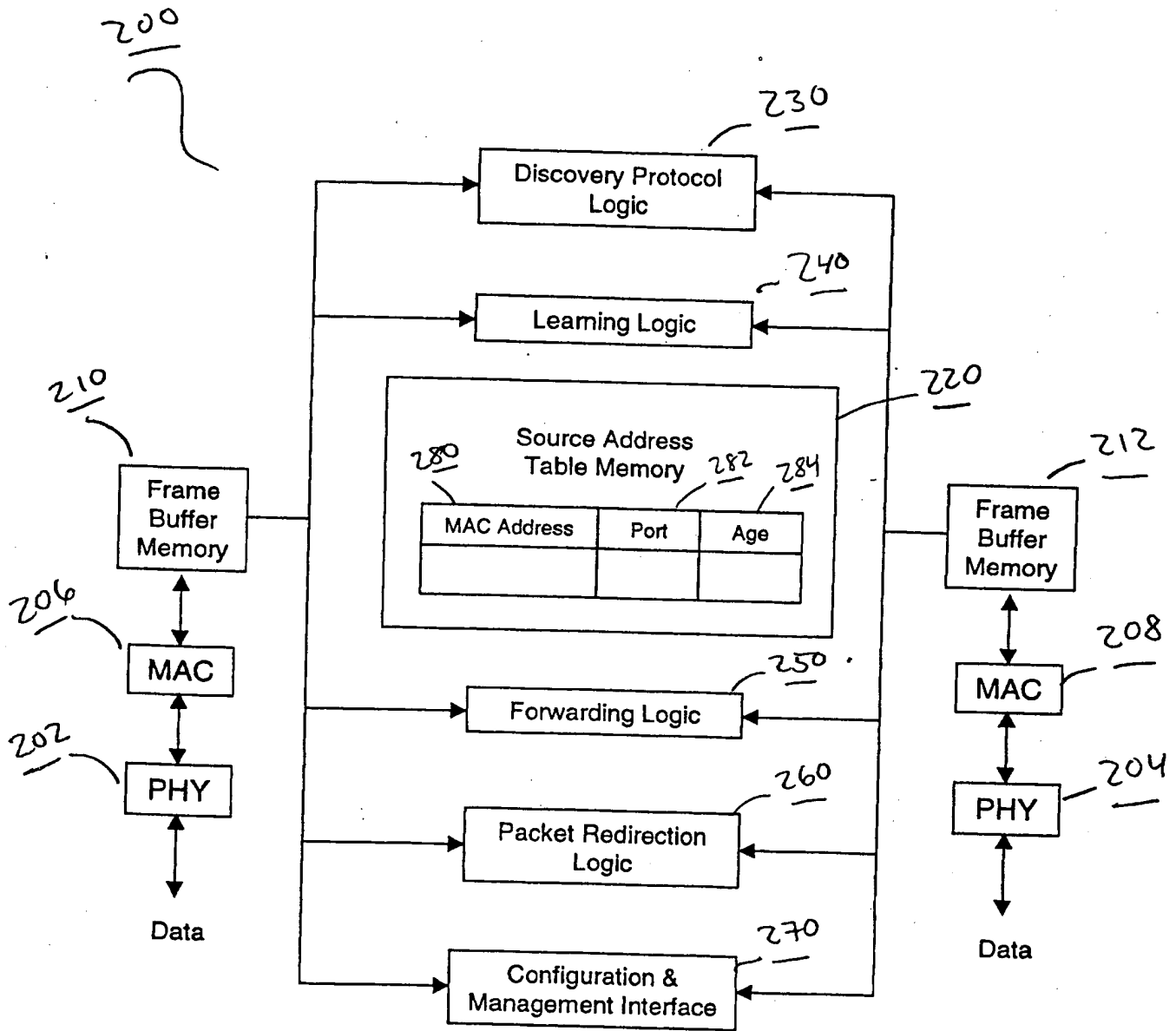


FIG. 2-A

PRIOR ART



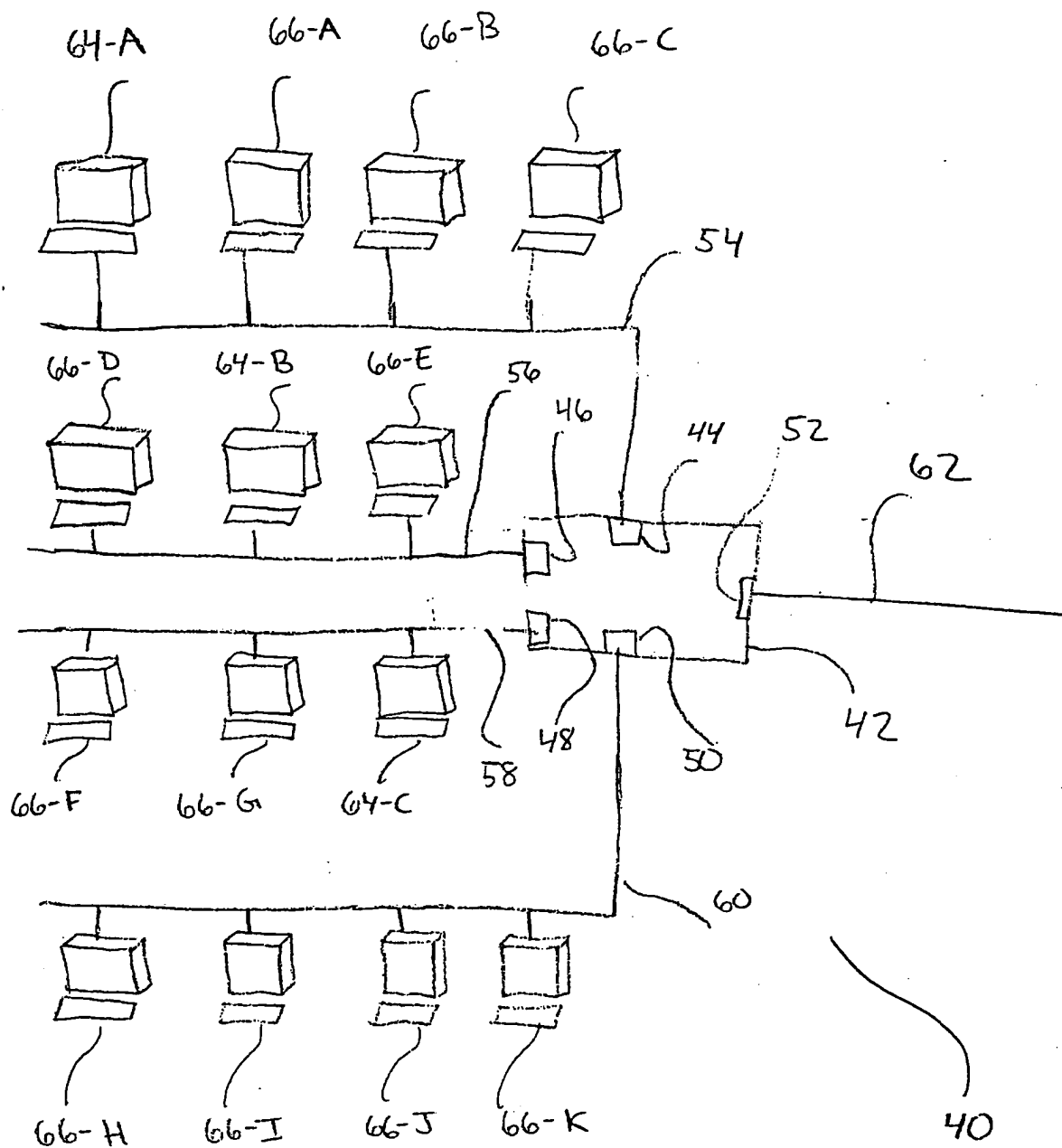


FIG. 3

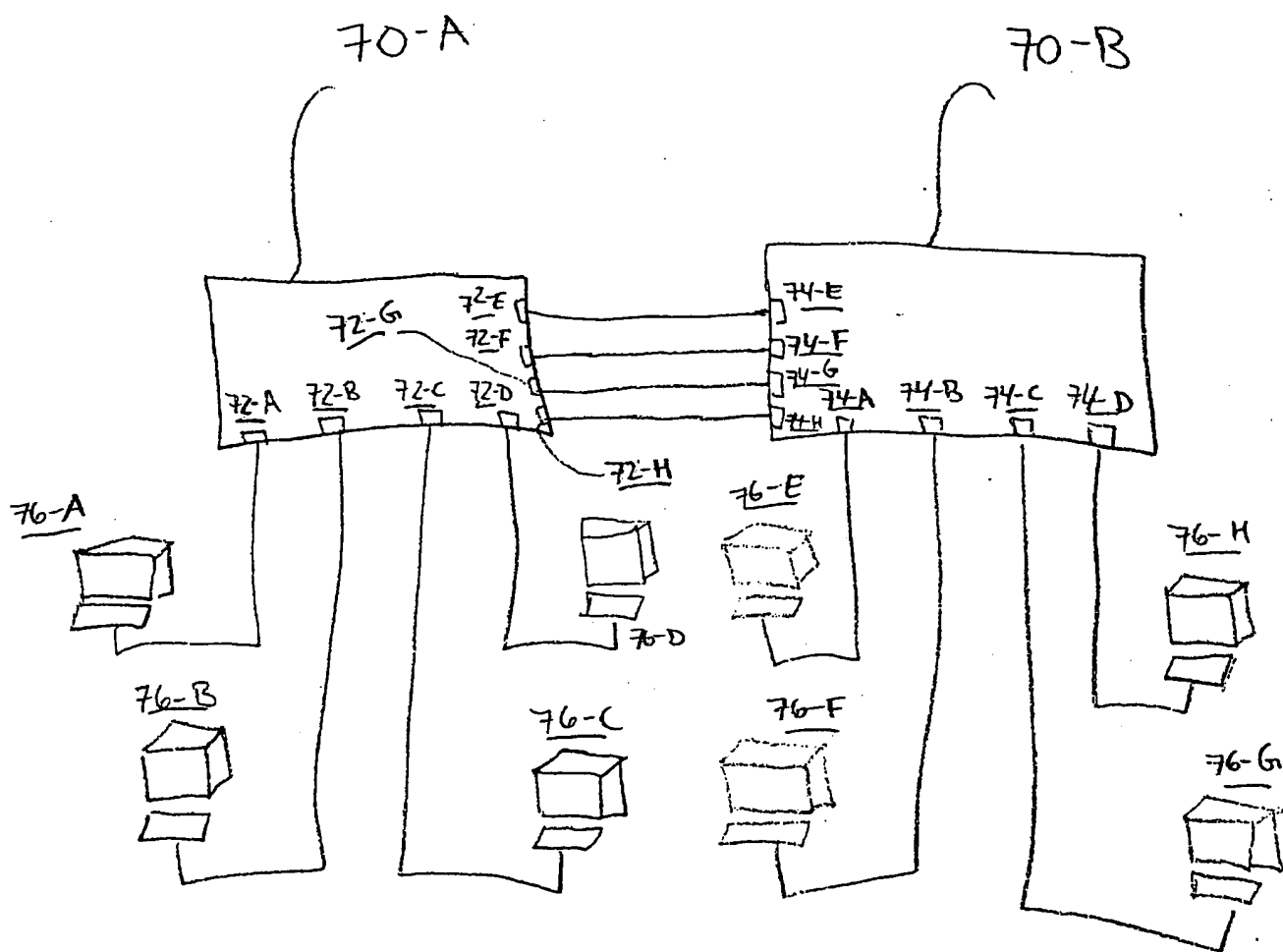


FIG. 4

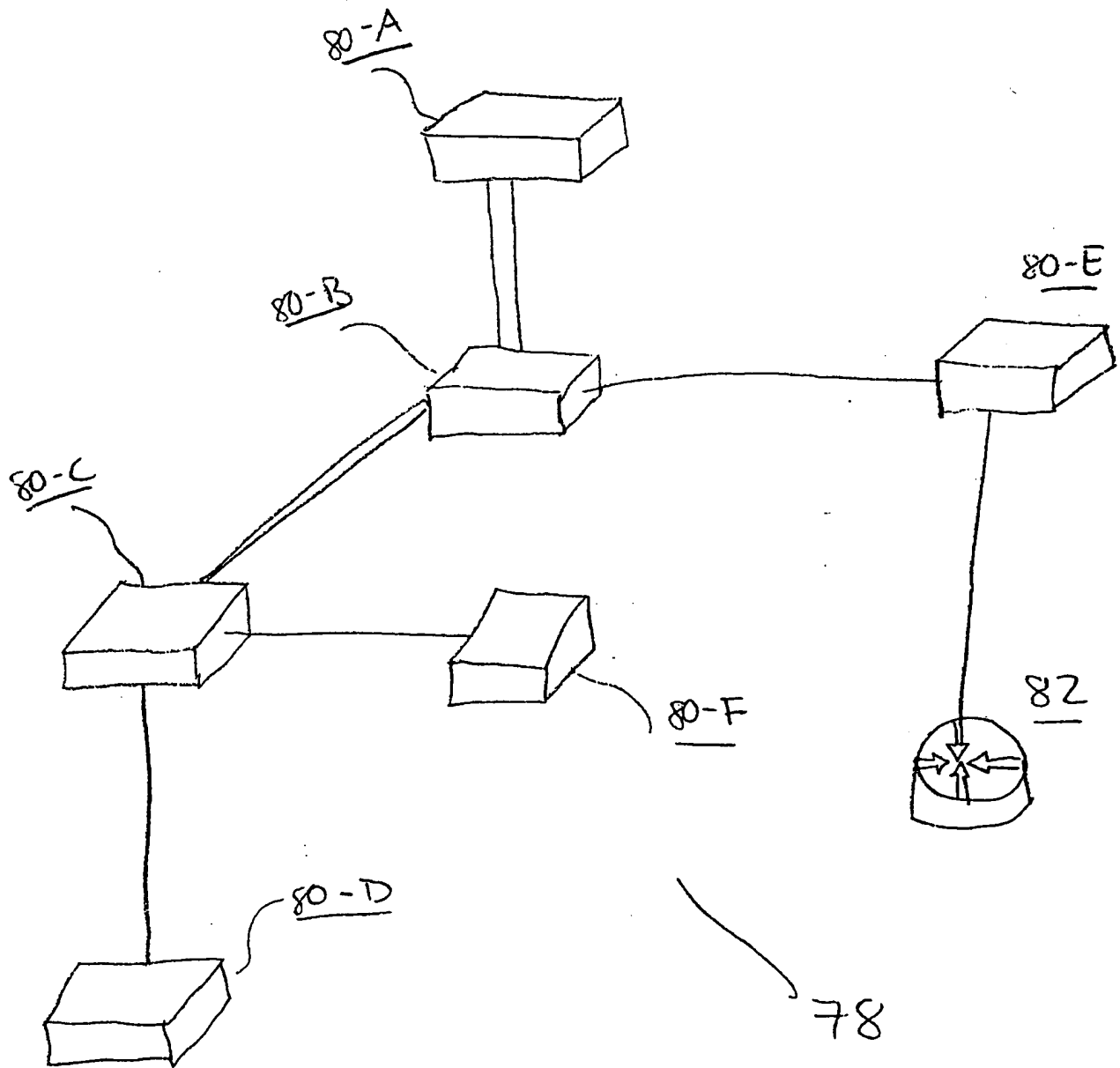
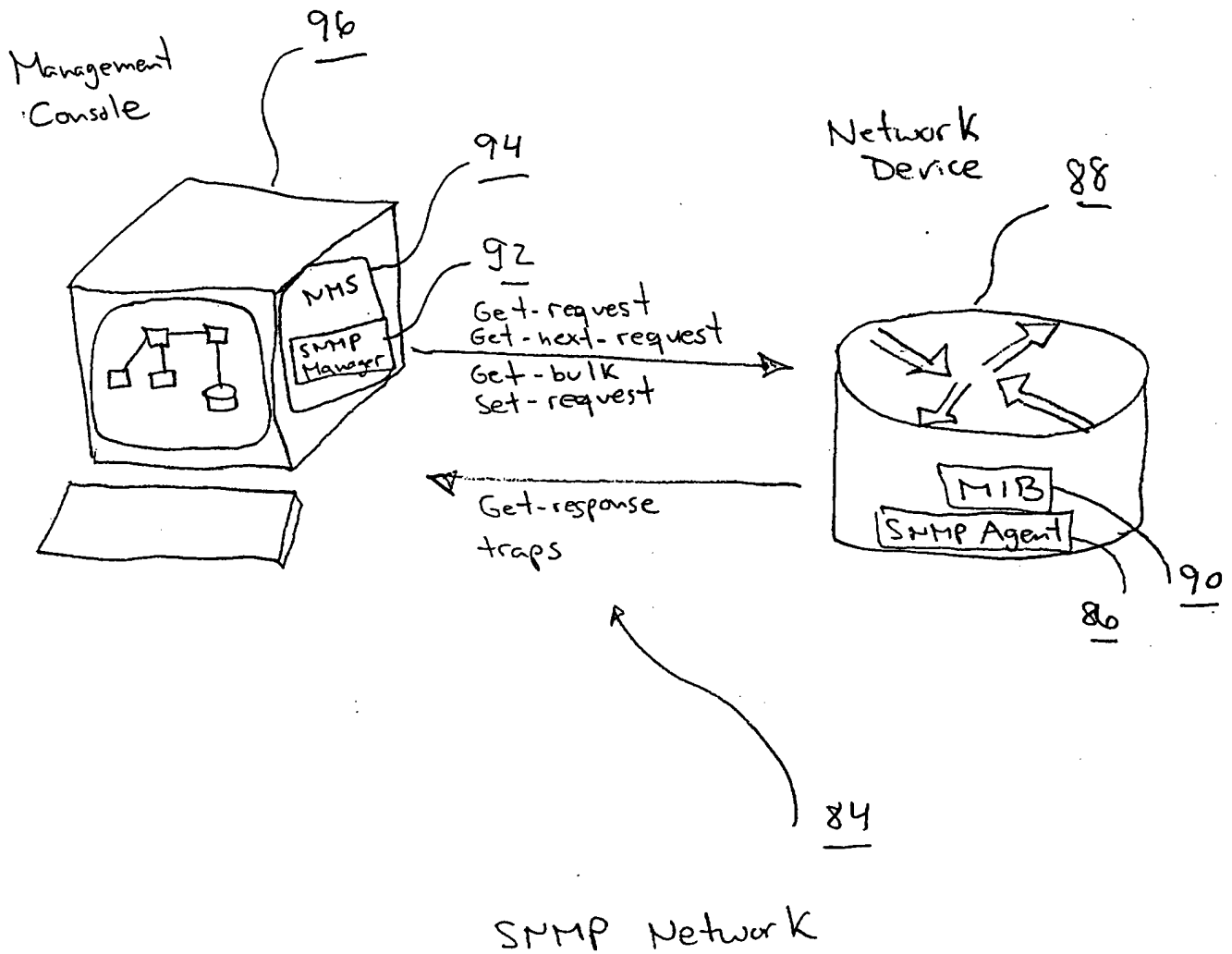


FIG. 5



PRIOR ART

FIG. 6

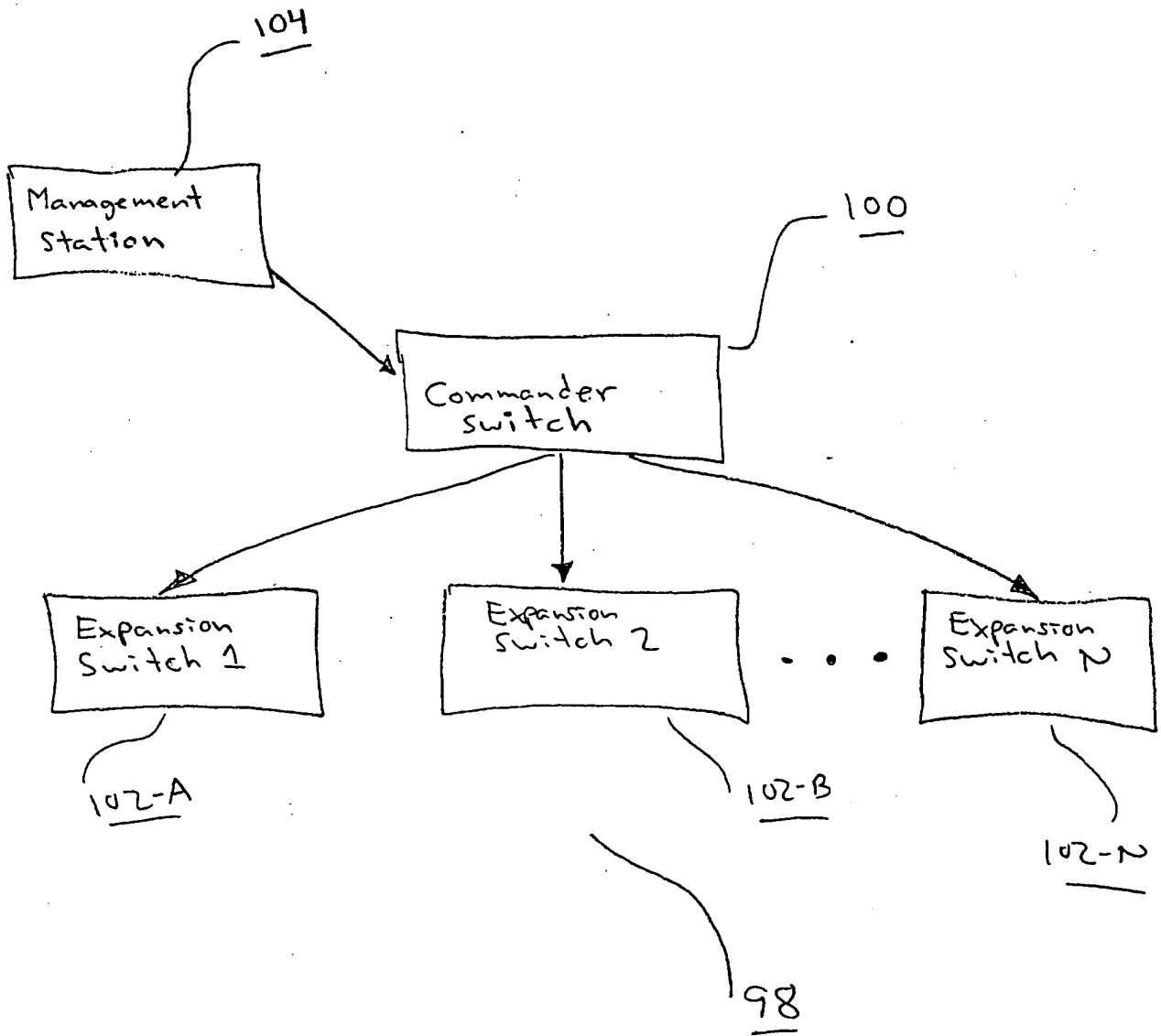


FIG. 7



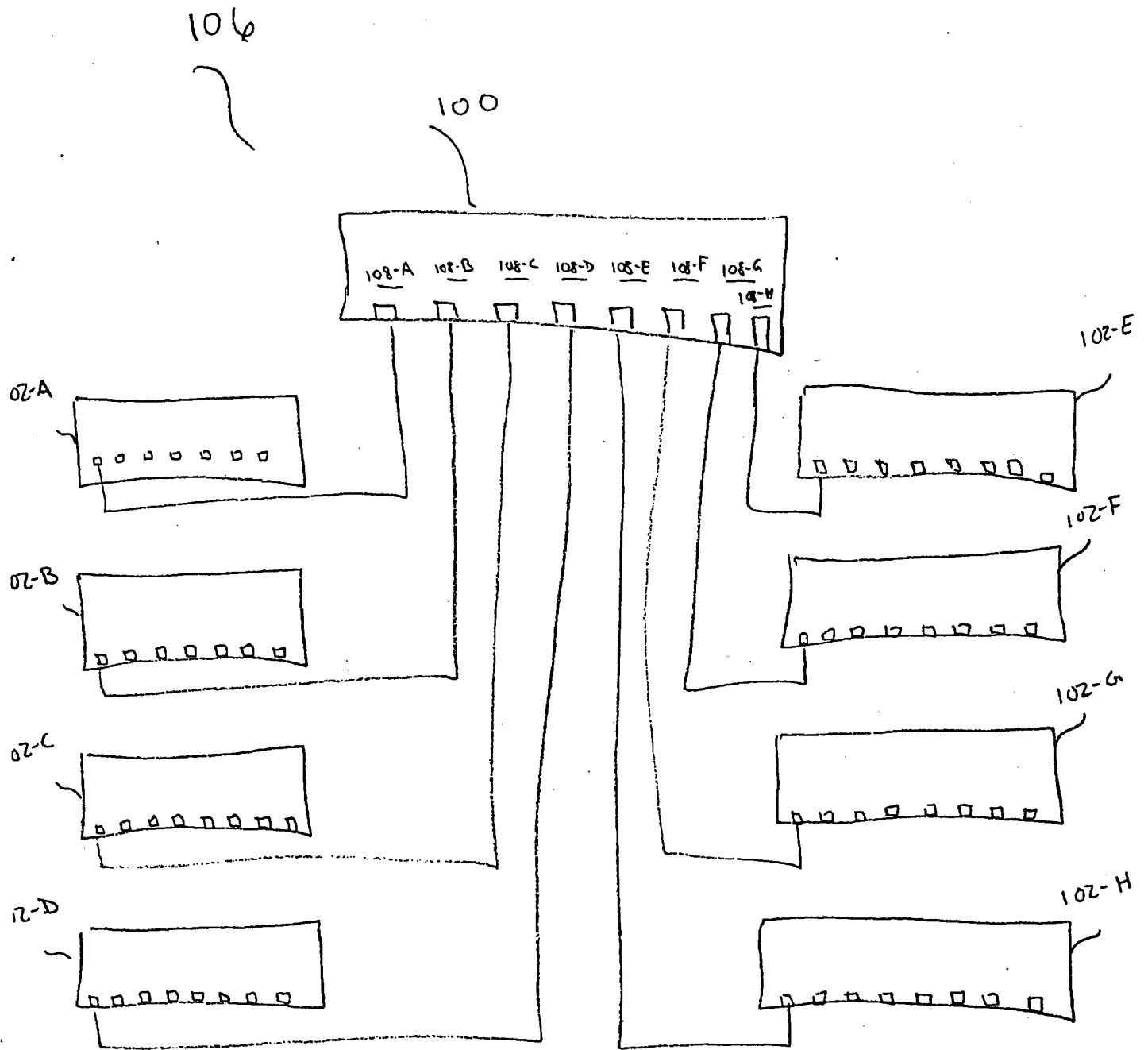


FIG. 8

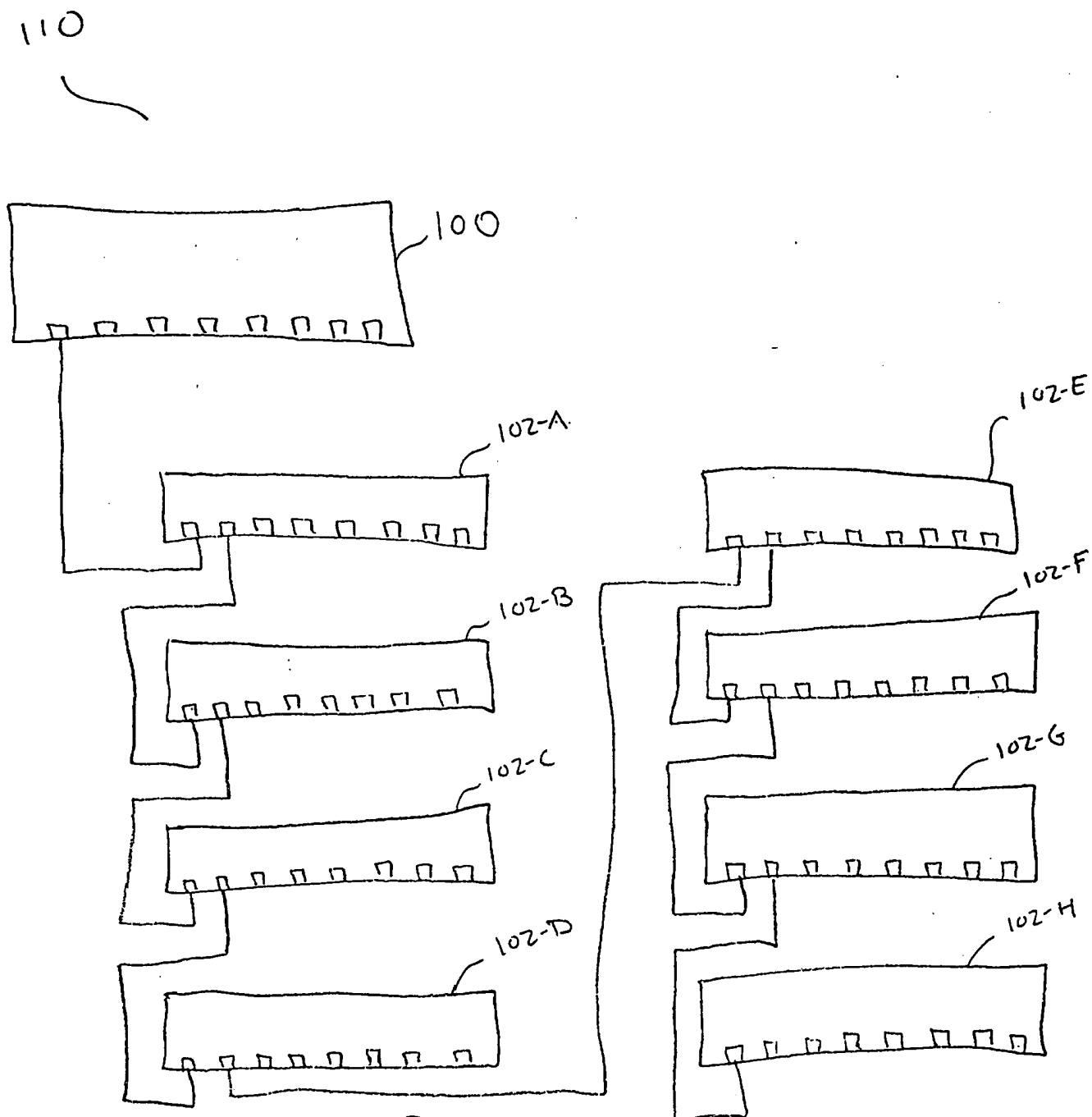


FIG. 9

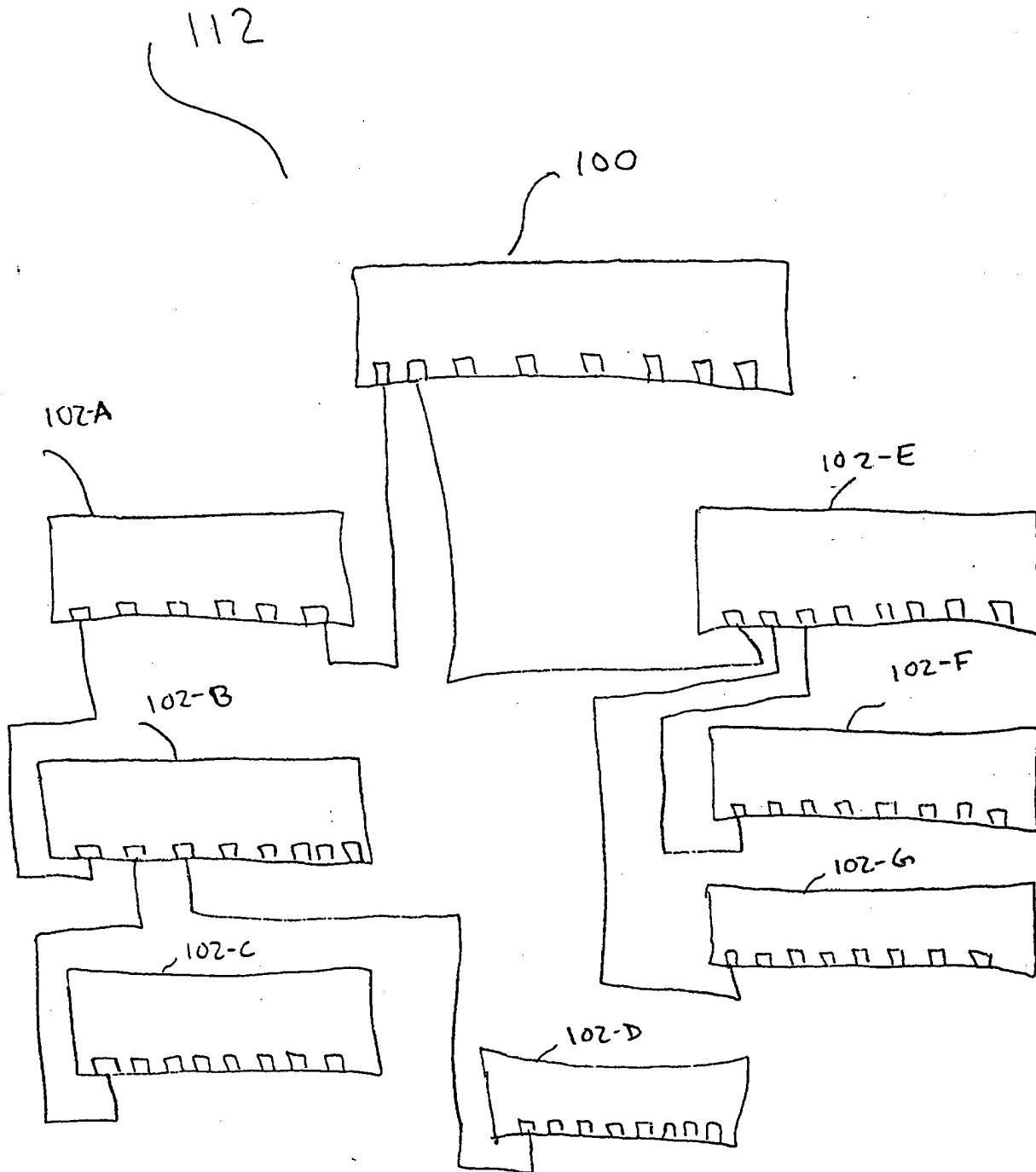


FIG. 10

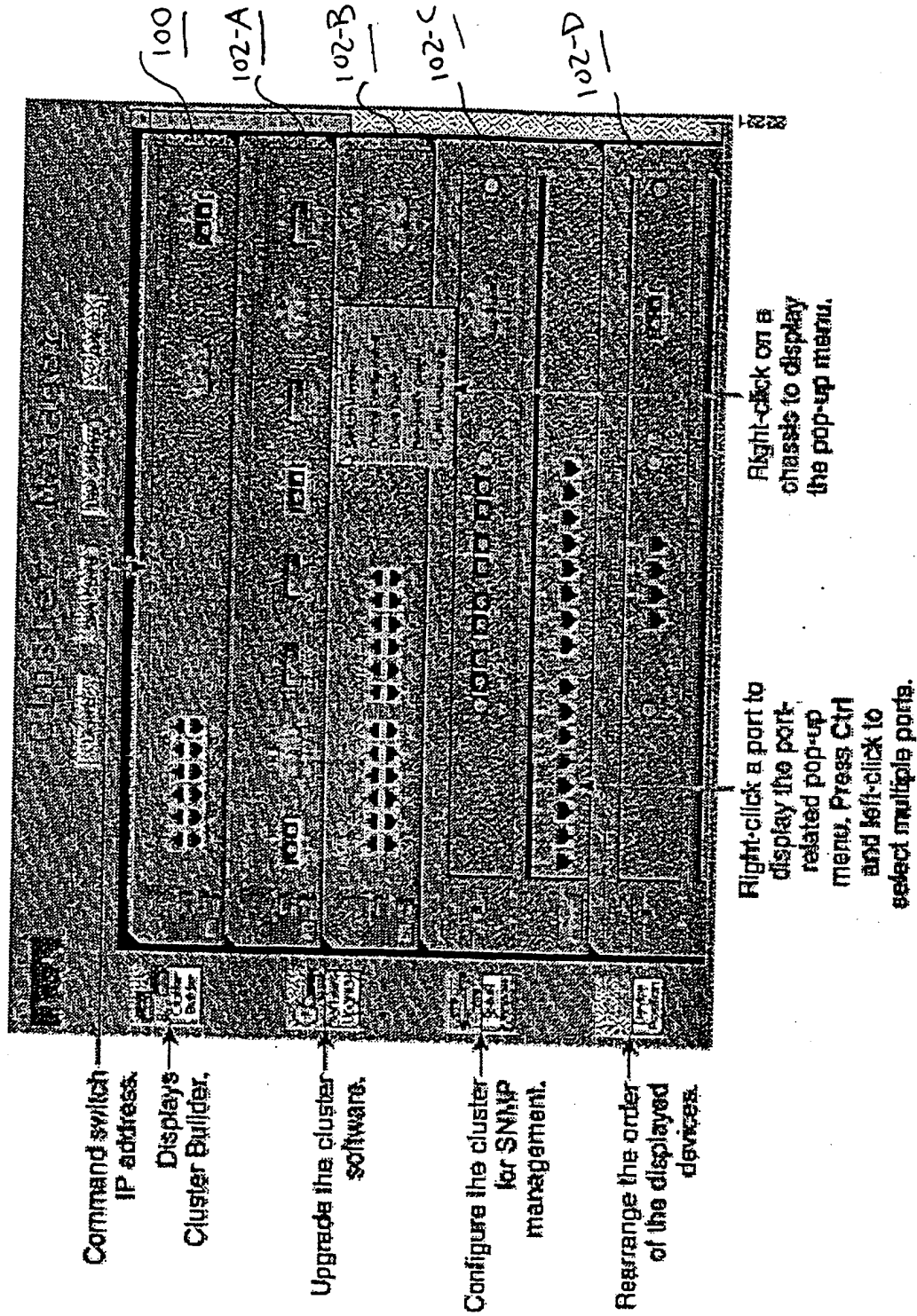


FIG. 11

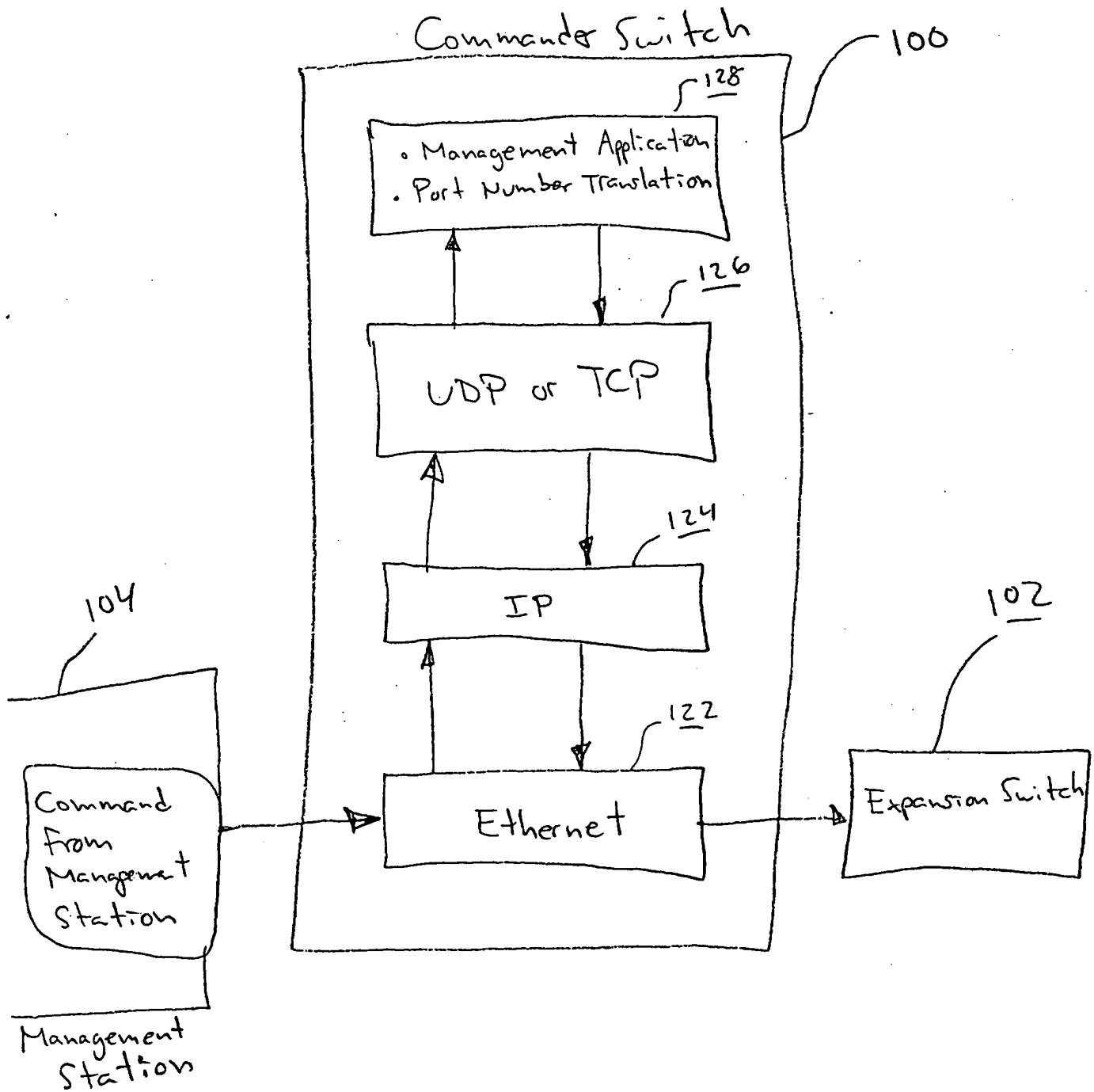


FIG. 12

### *cluster\_member* data structure

```
typedef struct cluster_member_  
{  
    ushort mbr_num;          /* Number assigned to this member (0xFFFF if  
                             non-member) */  
    ushort tp_flags;         /* CMP Transport flags for this member */  
    uchar mbr_flags;         /* Active - 0x01, Inactive 0x00 */  
    uchar mbr_heartbeat;     /* 0 each time hb is rcvd from cmdr; increment  
                             each time a heartbeat is sent. Should be 0  
                             or 1 most of the time. */  
    uchar mbr_numhops;       /* # CDP hops from commander */  
    uchar mbr_update;        /* Counter for automatic neighbor updates */  
    ipadrtype mbr_cmpaddr;    /* assigned CMP address */  
    ipadrtype mbr_ipaddr;     /* regular IP address */  
    ipadrtype cmdr_ipaddr;    /* IP address of member's commander */  
    ipadrtype cmdr_cmpaddr;   /* CMP address assigned to commander */  
    uchar cmdr_macaddr[ IEEEBYTES ]; /* MAC addr of member's commander */  
    uchar mbr_macaddr[ IEEEBYTES ]; /* store instead of get each time */  
    uchar mbr_CmdPortID[ CDP_MAX_PORT_ID_STR_LEN ];  
    uchar platform_name[ CDP_MAX_PLATFORM_STR_LEN ];  
    uchar host_name[ MAX_HOST_NAME ];  
    uchar cluster_name[ MAX_CLUSTER_NAME ];  
} cluster_member;
```

FIG. 13

## *cluster\_neighbor* data structure

```
typedef struct cluster_neighbor_ {  
    ipaddrtype cn_cmpaddr;          /* IP address assigned to the neighbor */  
    ipaddrtype cn_ipaddr;          /* Neighbors assigned IP address */  
    ulong cn_capabilities;         /* reg_invoke_cdp_lookup_cache_info_1 */  
    ushort cn_mbrnum;             /* 0xFFFF if not a stack member */  
    uchar cn_qualification;        /* 0 Qualified  
                                   1 Not cluster capable (no CMP hello)  
                                   2 Not cluster capable (ver mismatch)  
                                   3 cluster capable; STP-BLK at either end  
                                   4 Belongs to a different stack */  
    uchar cn_qualification_note;   /* 0x01 Has Configured IP address  
                                   0x02 Saw > 1 CDP neighbor, not point-to-point  
                                   0x04 Sender port is Fast Ethernet */  
    uchar cn_sender_numhops;      /* 0-MAX_CLUSTER_SIZE, number of CDP hops to  
                                   commander */  
    uchar cn_mbr_fec_number;      /* 0 if single port, 1-MAX_FEC if grpd */  
    uchar cn_fec_number;         /* 0 if single port, 1-MAX_FEC if grpd */  
    uchar cn_pad;                /* Keep it even */  
    uchar cn_macaddr[IEEEBYTES]; /* MAC address of CDP sender */  
    uchar cn_port_macaddr[IEEEBYTES]; /* MAC address of CDP sender port */  
    uchar cn_cmdr_macaddr[IEEEBYTES]; /* MAC addr of sender's commander */  
    uchar cn_portID[CDP_MAX_PORT_ID_STR_LEN]; /* CDP sender Port ID */  
    uchar cn_mbr_portID[CDP_MAX_PORT_ID_STR_LEN]; /* Receiving port's ID */  
    uchar cn_hostname[MAX_HOST_NAME]; /* Sender platform str */  
    uchar cn_platform_name[CDP_MAX_PLATFORM_STR_LEN];  
} cluster_neighbor;
```

FIG. 14

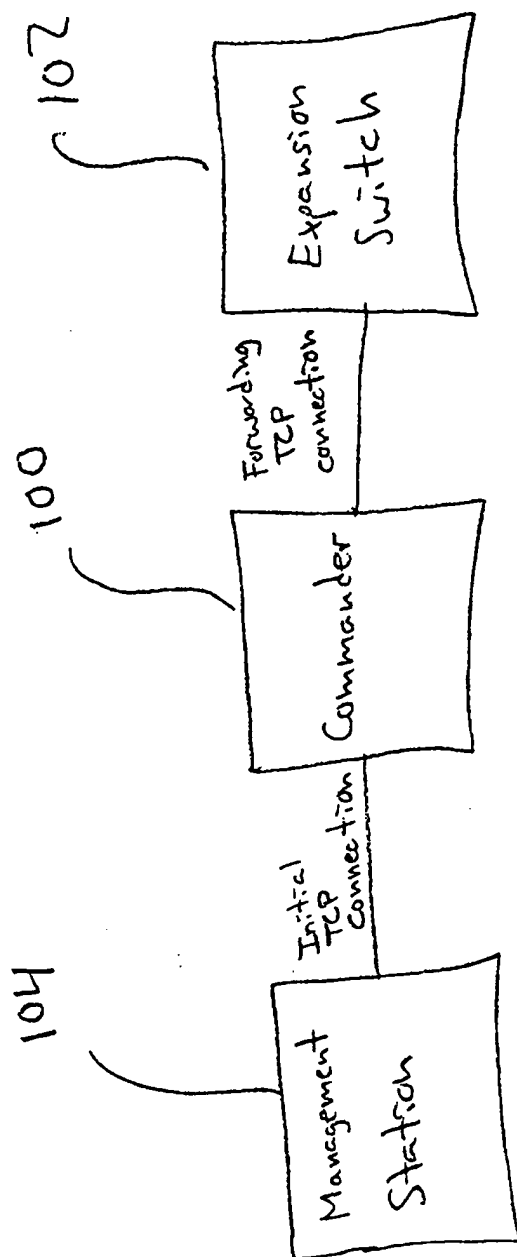


FIG. 15